Patent Application of

Geoff BARRETT
28 Devonshire Road
Westbury Park, Bristol, BS6 7NJ
GREAT BRITAIN

for

APPARATUS FOR PROVING SYSTEM PROPERTIES

METHOD AND APPARATUS FOR PROVING SYSTEM PROPERTIES

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to a method and apparatus for reducing the complexity of a representation of a hardware system.

DESCRIPTION OF THE PRIOR ART

The first stage in synthesizing and proving the properties of a system is a compilation process in which the system is modelled by representation as a set of functions comprising:-

a first subset of functions which determines the value of system outputs as a function of system inputs, system states represented by state bits, and internal signals;

a second subset of functions which determines the values of state bits on the next clock cycle as a function of system inputs, system states represented by state bits, and internal signals; and

a third subset of functions which determines the values of internal signals as a function of system inputs, system states, and internal signals.

To enable or accelerate formal proof of the system and its properties, internal signals may be eliminated from the system model by substituting them into the functions which refer to them. In the course of this substitution, the representation of the model may become extremely large. If this occurs, it is possible to detect an explosion in the size of the representation and to suspend the substitution process while restructuring the representation to seek a reduction in size.

Typically in a compilation process, static relationships between signals in the system model can be destroyed by dynamic restructuring operations. This can lead to a further explosion later during the substitution process.

It would be advantageous to take static relationships into account during the dynamic restructuring process.

One technique of representing functions and internal signals is by the use of binary decision diagrams (BDD's). A binary decision diagram is a representation of a digital function which contains the information necessary to implement the function. The diagram is a tree-like structure having a root and plural nodes, where the root represents the digital function and the nodes are labelled with variables. Each node has two branches, one representing the assertion that the variable labelling the node is 1, and the other representing the assertion that the variable labelling the node is 0. In a BDD, "ordering" relates to the order in which variable names are encountered during traversal of the graph. Better orderings result in fewer nodes in the graph.

SUMMARY OF THE INVENTION

According to a first aspect of the present invention, there is provided a method for selecting an order in which to sift variables in a binary decision diagram comprising:-

arranging the variables of a binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node, correspond to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth first manner, thereby to produce a list of said labels in said selected order.

According to a second aspect of the present invention there is provided apparatus for selecting an order in which to sift variables in a binary decision diagram comprising a first store storing bits representing the variables of the binary decision diagram;

a second store; and

a processor adapted to arrange the said variables of said

binary decision diagram in a representation of the nodes of a graph in which the nodes are labelled with the variables such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and to

traverse the graph in a depth-first manner such that said processor means outputs to said second store a list of said labels in said selected order.

According to a third aspect of the present invention there is provided a method for restructuring a binary decision diagram representative of a hardware system, comprising:-

arranging the variables of a binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth-first manner to produce a list of said labels in a selected order;

using said selected order, controlling sifting each variable.

Preferably said variables are sifted one-by-one to a deepest best location. Advantageously said variables are sifted one-by-one in said selected order to a deepest best location followed by sifting in reverse order to a shallowest best location.

According to a fourth aspect of the present invention there is provided apparatus for restructuring a binary decision diagram comprising:-

storage circuitry for storing bits representative of a set of functions as binary decision diagrams having a plurality of nodes labelled by variables;

a processor for detecting a number of nodes of said binary decision diagram, and in response to such detection, arranging the variables of said binary decision diagram on the nodes of a graph in which the nodes are labelled such that the set of

functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node, traversing the graph in a depth-first fashion to produce a list of labels in a selected order and using said selected order, controlling sifting of variables of said binary decision diagrams;

wherein said sifted binary decision diagram is written by said processor to said storage.

According to a fifth aspect of the present invention there is provided a method for proving the properties of a hardware system comprising:-

representing said system as binary decision diagrams having a plurality of nodes labelled by variables;

substituting functions which determine variables of internal signals;

arranging the variables of a binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth-first manner to produce a list of said labels in a selected order;

using said selected order, controlling sifting each variable.

According to a sixth aspect of the present invention there is provided apparatus for proving the properties of a hardware system comprising:

storage circuitry for storing bits representative of a set of functions which represent the hardware system as binary decision diagrams having a plurality of nodes labelled by variables;

processor means for substituting functions which determine the values of internal signals into the set of functions representing said system and detecting an increase in the number of nodes of said binary decision diagram, and, in response to

such detection arranging the variable of said binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponding to the set of functions which depend on the variables labelling the node, traversing the graph in a depth-first fashion to produce a list of labels in said selected order, and using said selected order controlling sifting of the variables of said binary decision diagram; and

further comprising a second store, wherein said sifting binary decision diagram is written by said processor to said second store.

Preferably said number is a threshold derived from an original number of nodes.

Alternatively said number of nodes is the number of nodes which branches on a predetermined variable.

Alternatively said number is an absolute number.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the present invention will now be described with respect to the following drawings in which:-

Figure 1 shows a binary decision diagram for the function

$$f = x \text{ OR } y;$$

Figure 2 shows a logical diagram of a multiplexer;

Figure 3 shows a binary decision diagram for the equation;

$$b_i = \text{NOT } (a_i \text{ AND } s_i);$$

Figure 4 shows the binary decision diagram for the equation

$$d = \text{NOT } (b_1 \text{ AND } b_2 \text{ AND } b_3 \ldots b_n);$$

Figure 5 is an optimally ordered substitution of the equations of Figures 3 and 4;

Figure 6 shows a graph of relationships between the variables of the multiplexer of Figure 2; and

Figurer 7 shows an example of apparatus arranged to implement an embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A Binary Decision Diagram (hereinafter referred to as a BBD) is a directed acyclic graph representative of a Boolean function as a decision procedure based on the variables on which it depends. For instance, for the function:-

f = x OR y,

f can be implemented by the decision procedure "if x then true else if y then true else false". Each of the "if...then...else..." constructs of this decision procedure can be represented as a node in a graph.

Referring to Figure 1, the first node 1 is labelled with the variable x and there are two branches from this first node, one 11 is "true" and the other 12 is "if y then true else false". This other branch 12 leads to a second node 2 which is labelled with the variable y, which in turn has two branches 21, 22 of which one is "true" and the other is "false".

It will be understood that although the nodes 1 and 2 are described above as being labelled with variables, nevertheless these labels could in fact refer to functions which upon evaluation would give rise to the logical values "true" or "false".

Referring now to Figure 2, a multiplexer consists of a first set of n NAND gates 10_1-10_n, each gate having two respective inputs a_1-a_n, s_1-s_n. The outputs lines b_1 and b_n of the gates are connected to an n-input NAND gate 20 having an output d.

Thus, in terms of a system as described in the preamble to this patent application, the multiplexer of Figure 2 has system inputs (a_1-a_n, s_1-s_n), internal signals (b_1-b_n) and a system output (d). The output d is related to the internal signals b_1-b_n by the equation:-

$$\mathbf{d = NOT\ (b_1\ AND\ b_2\ AND\ b_3 \ldots b_n)}$$

and each internal signal b_i to the respective inputs a_i and s_i by the equation

$$\mathbf{b_i = NOT\ (a_i\ AND\ s_i)}$$

Thus,

$$\mathbf{d = (a_1\ AND\ s_1)\ OR\ (a_2\ AND\ s_2)\ OR \ldots (a_n\ AND\ s_n)}$$

Referring to Figure 3 the relationship b_i = NOT (a_i AND s_i) is shown as a binary decision diagram.

Figure 4 shows the binary decision diagram representation of the expression for d in terms of the internal signals b.

By inspection, there are 3n variables (a_i, s_i and b_i) and there are thus (3n)! apparently equally good orderings possible. However, by inspection of the overall equation for the device it would be seen that a_1 and s_1 are associated together, a_2 and s_2 are associated together and so on which means that there are in fact only n! orderings which are optimal for the entire system.

An advantage of the present invention is that it enables more information about the system as a whole to be taken into account when performing operations which would otherwise not take this information into account. Failing to take the information into account can result in following paths which do not lead to a solution, or which are highly inefficient in reaching the solution.

Figure 5 shows a binary decision diagram for the multiplexer of Figure 2 in which the respective pairs of inputs are associated together.

The size of a binary decision diagram is sensitive to the order in which the variables are inspected, and efficient BDD reordering is very important. One algorithm for reordering is "sifting", wherein each variable is taken in turn and the best position of it is found by trying it in every possible position of the BDD. It is then necessary to decide which variable to take first. A known and frequently successful tool for doing this is to rank the variables according to which variable labels the greatest number of nodes and then to sift in the order of ranking.

In the present BDD, it is clear that each variable labels a single node and thus it would not be possible using known techniques to identify a highest ranking variable. Conventionally, in such a situation, an arbitrary order for sifting would be used.

The present invention makes use of a function graph which is traversed to determine an order for sifting.

As used herein, a function graph is a directed acyclic graph where the leaves are labelled with functions and the nodes are labelled with sets of variables (non-empty). The only restriction put on this graph is that a variable which is in the set labelling a node is in the "cone" of all the functions at the leaves below it and no others. This restriction plus the fact that the sets of variables must be non-empty, is enough to ensure that the graph is unique. The "cone" of a function is herein defined to be all those variables on which a function depends, either directly or through the intermediate signals on which it depends.

Using a function graph to define an ordering of the

variables in a BDD to minimise its size may be related to the register allocation technique used in software compilation in that the ordering of the variables is derived from a traversal of the function graph in such a way that no node is visited before all of its predecessors has been visited, but each node is visited as soon as all its predecessors have been visited, unless there is a race between more than one node, in which case one of the competing nodes is chosen and its subgraphs traversed first.

Figure 6 shows a function graph for the multiplexer of Figure 2 having a root labelled by b_1-b_n intermediate nodes labelled by a_1, s_1, a_2, $s_2 \ldots a_n$ s_n and leaves as shown. Traversing this function graph from the top down gives the order:-

b_1, $b_2 \ldots b_n$, a_1, s_1, a_2, $s_2 \ldots . a_n$, s_n

By using this order which is derived from static information of the system, the binary decision diagrams of (in this case) Figures 1, 3 and 4 are sifted to provide an optimal order. This order is that represented by Figure 5.

It should be noted that substitution may be effected without restructuring the BDD, while monitoring the size of the BDD. If an explosion in BDD size is detected, sifting is then effected on the basis of the order provided by the present invention.

Figure 7 shows an example of apparatus aranged to implement an embodiment of the present invention. A first storage circuitry 30 stores data representative of the variables of a binary decision diagram.

The first storage circuitry has an input 32 for receiving the variables. The output 34 of the first storage circuitry is coupled to an input of a processor 40. The processor receives the variables via the output 34 of the first storage circuitry and consults a function graph by arranging the variables in a representation of the nodes of the graph such that the nodes are

labelled with the variables so that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node, as shown in Figure 6. The processor then traverses the graph in a depth-first manner, as indicated in Figure 6 by the arrow, to construct a list of the labels in a selected order. The list is output from the processor via output 44 which is coupled the input 52 of second storage circuitry.

While the invention has been previously shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.